

**REMARKS**

This amendment is filed in response to the Office Action mailed on November 3, 2004. All objections and rejections are respectfully traversed.

Claims 1-19 are in the case.

No Claims were amended.

Claims 20-33 were added to better claim the invention.

At Paragraph 1 of the Office Action the incorporation by reference was objected to. Amendment of the Specification is believed to satisfy this objection.

At Paragraphs 2- 4 claim 1 was rejected under 35 U.S.C. § 102 (e) as being anticipated by Madnick U. S. Patent No. 6,282,537 issued August 28, 2001 (hereinafter Madnick).

Applicant's claimed invention, as set forth in Claim 1, comprises in part:

1. A method for converting a file access data structure from a first endianness to a second endianness, the method comprising the steps of:  
*identifying, from a descriptor look up table, a series of actions to perform on elements of the file access data structure; and*  
*performing the identified series of actions on the elements of the file access data structure.*

Madnick discloses using a structured query language to query both structured documents and to query semi-structured documents. Semi-structured documents may be, for example, a web page written in HTML language. The structured query language requests particular data, and the semi-structured documents are searched by comparing contexts in the documents with desired contexts. Data found in a desired context is transmitted to a translator which places the data in a structured format.

Applicant respectfully urges that Madnick has no disclosure of Applicant's claimed novel *identifying, from a descriptor look up table, a series of actions to perform on elements of the file access data structure; and*

*performing the identified series of actions on the elements of the file access data structure.*

In particular, Madnick has no disclosure of *a descriptor look up table*. A *descriptor look up table* is defined in Applicant's Specification as related to a file access data

structure. A file access data structure is defined in the Specification at Page 12 Lines 6-12 as:

Fig. 4 is a flow chart detailing the process that the DAFS layer 318 performs upon receipt of a file access data structure, such as a DAFS data structure, from a client. By “file access data structure” it is meant data structures associated with structure-based networking or file access protocols, such as the Direct Access File System, CIFS or NFS or other protocols in which packets are sent/received in non-native byte order. File access data structures do not include the various data structures associated with stream-based data such as that utilized by the Hypertext Transport Protocol (HTTP). In step 405 the client receives the DAFS data structure. This receipt is accomplished via known data transfer techniques such as virtual interface data connections.  
(Specification, Page 12, lines 6-12)

The descriptor look up table contains information to direct the method for converting a file access data structure from a first endianness to a second endianness. This function of the descriptor lookup table is set out in the Specification at Page 7 lines 1-7 as:

Within a storage operating system, the byte-swapping engine may be called by various processes to translate a data structure into a proper endianness. The calling process places the data structure in an input buffer and then calls the byte-swapping engine. After the byte-swapping engine has performed the actions described in the descriptor look-up table, the byte-swapped data structure is placed in an output buffer. The calling process can then access the byte-swapped data structure from the output buffer for further processing.

Specification Page 7 lines 1-7)

Applicant respectfully urges that Applicant's claimed *identifying, from a descriptor look up table, a series of actions to perform on elements of the file access data structure; and*

*performing the identified series of actions on the elements of the file access data structure*, is totally absent from the disclosure of Madnick. That is, Madnick analyzes semi-structured documents to obtain data from them, where in sharp contrast, Applicant claims using a *descriptor look up table* to analyze Applicant's *file access data structure*, and ultimately to perform byte swapping on Applicant's *file access data structure*.

Therefore, Applicant respectfully urges that Madnick is legally precluded from anticipating Applicant's claimed invention under 35 U.S.C. § 102 because of the absence from the disclosure of Applicant's claimed novel *identifying, from a descriptor look up table, a series of actions to perform on elements of the file access data structure; and performing the identified series of actions on the elements of the file access data structure*.

Further, Applicant's claimed invention is for "converting a file access data structure from a first endianness to a second endianness".

Applicant further respectfully urges that Madnick has no disclosure of converting the endianness of a file from one form to another, as claimed by Applicant. That is, Applicant respectfully urges that the absence from Madnick of any mention of the endian-

ness used in a file access data structure precludes Madnick from anticipating Applicant's claimed invention under 35 U.S.C. § 102 (e).

At Paragraphs 5-8 of the Office Action Claims 16-18 were rejected under 35 U.S.C. § 102 (e) as being anticipated by Bowman-Amuah U. S. Patent No. 6,434,568 issued August 13, 2002 (hereinafter Bowman-Amuah).

Applicant's invention, as set forth in representative claim 16, comprises in part:

16. A method for converting elements of a file access data structure from a first endianness to a second endianness, the method comprising the steps of:

*determining a type of the file access data structure;*  
*processing, in response to the file access data structure of being of a first type, the file access data structure along a first processing path;*  
*processing, in response to the file access data structure being of a second type, the file access data structure along a second processing path.*

Bowman-Amuah discloses a communication system which uses a client system using a new format to communicate with a legacy system. The new format, for example the new format may be a web page and a browser to access the web page, contacts a legacy system using an object designed to interface between the new system and the legacy system. The legacy system may be a business oriented software package, and the object provides a communication link between the two systems, where the two systems are the

new format running on the client and the legacy system. The object is designed to specifically provide communication between the particular new system and the particular legacy system.

Applicant respectfully urges that Bowman-Amuah has no disclosure of translating a first endianness to a second endianness in movement of data between systems. In particular, Applicant respectfully urges that Bowman-Amuah has no disclosure of Applicant's claimed novel:

*determining a type of the file access data structure;*  
*processing, in response to the file access data structure of being of a first type,*  
*the file access data structure along a first processing path;*  
*processing, in response to the file access data structure being of a second type,*  
*the file access data structure along a second processing path.*

Bowman-Amuah simply uses his object to perform communication between his client, using a new format, and a legacy system. Applicant, as set out in representative claim 16, first does *determining a type of the file access data structure*, and after doing the determining step, *processing, in response to the file access data structure of being of a first type, the file access data structure along a first processing path* and alternatively, if the determining step finds the file access data structure to be of a second type, *processing, in response to the file access data structure being of a second type, the file access data structure along a second processing path.*

Accordingly, Applicant respectfully urges that Bowman-Amuah is legally precluded from anticipating Applicant's claimed novel invention under 35 U.S.C. § 102 (e) because of the absence from the disclosure of Bowman-Amuah of Applicant's claimed novel:

*determining a type of the file access data structure;*

*processing, in response to the file access data structure of being of a first type, the file access data structure along a first processing path;*

*processing, in response to the file access data structure being of a second type, the file access data structure along a second processing path.*

At Paragraphs 9-13 of the Office Action, claims 2, 11, and 15 were rejected under 35 U.S.C. § 103 (a) as being unpatentable over Bowman-Amuah and Lee et al. U. S. Patent No. 5,867,690 issued February 2, 1999 (hereinafter Lee).

Applicant's claimed invention, as set forth in representative Claim 2, comprises in part:

2. A method of converting elements of a file access data structure from a first endianness to a second endianness, the method comprising the steps of:

determining if the file access data structure is a critical path data structure;

converting, in response to the file access data structure *being a critical path data structure, the elements from the first endianness to the second endianness using a set of specific code functions;*

converting, in response to the file access data structure not being a critical path data structure, a header of the file access data structure from the first endianness to the second endianness using a second set of specific code functions; and

*calling a byte swapping engine to convert selected elements of the file access data structure from the first byte order to the second byte order.*

Lee discloses system to convert data from one endian format to another endian format, where his system has a hardware byte swapping device connected between a processor which uses one endian format and a data storage system which uses a second endian format. The hardware device swaps bytes as a data stream passes through the device. A port of the hardware byte swapping device may be chosen to provide a desired form of byte swapping. Alternatively, Lee's processor controls the hardware byte swapping device by a control bus which is activated by signals from the processor.

Applicant respectfully urges that Lee has no disclosure of Applicant's claimed novel determining if the "file access data structure" is a *critical path data structure*. A critical path data structure is explained in Applicant's Specification Pages 6 Line 11-30:

"This invention overcomes the disadvantages of the prior art by providing a system and method for byte swapping file access data struc-

tures from a first endianness to a second endianness. The system and method first determines if a file access data structure contains certain characteristics. If the file access data structure contains one of these specific characteristics, the data structure is processed by a processing path using static programming techniques. These statically programmed techniques enable certain critical path or small data structures to be programmed more efficiently.

If the data structure is not one of the specific types to be processed using static programming techniques, the data structure is processed by a second processing path using a byte swapping engine located within a storage operating system executing on a computer. The byte-swapping engine utilizes a descriptor look-up table, which provides descriptions of the size of and action to be performed on each element of a file access data structure. The descriptor look-up table may also be nested by containing links to other descriptor tables within the descriptor look-up table. The byte-swapping engine works by stepping through the descriptor table and processing the elements of the desired data structure according to the element's size and action specified within the descriptor look-up table. The byte-swapping engine may be recursively called for nested entries of a file access data structure. By utilizing a table, wide variation in data structures can be accommodated and updates to data structures can be accomplished by modifying table entries." (Specification Pages 6 Line 11-30)

Accordingly, Applicant respectfully urges that Lee has no disclosure of Applicant's claimed novel:

converting, in response to the file access data structure *being a critical path data structure, the elements from the first endianness to the second endianness using a set of specific code functions;*

converting, in response to the file access data structure not being a critical path data structure, *a header of the file access data structure from the first endianness to the second endianness using a second set of specific code functions;* and

*calling a byte swapping engine to convert selected elements of the file access data structure from the first byte order to the second byte order.*

That is Lee simply has a hardware byte swapper which operates by a user selecting a port, or by a processor operating control signals on a control bus, and has no disclosure of Applicants use of whether or not the file access data structure is a *critical path* data structure to select different means for byte swapping, as claimed.

Accordingly Applicant respectfully urges that Lee and Bowman-Amuah, taken either singly or in combination, are legally precluded from rendering the presently claimed invention obvious under 35 U.S.C. § 103 because of the absence from both of Applicant's claimed novel:

converting, in response to the file access data structure *being a critical path data structure, the elements from the first endianness to the second endianness using a set of specific code functions;*

converting, in response to the file access data structure not being a critical path data structure, *a header of the file access data structure from the first endianness to the second endianness using a second set of specific code functions;* and

*calling a byte swapping engine to convert selected elements of the file access data structure from the first byte order to the second byte order.*

At Paragraphs 14 of the Office Action Claim 3 was rejected under 35. U.S.C. § 103(a) as being unpatentable over Bowman-Amuah and Lee, and further in view of Keele et al. U.S. Patent No. 5,438,674 (hereinafter Keele).

Applicant respectfully notes that claim 3 is dependent, and is dependent from an independent claim which is believed to be in condition for allowance. Accordingly, Claim 3 is believed to be in condition for allowance.

At Paragraphs 15 of the Office Action Claim 12 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bowman-Amuah and Lee in view of McCarthy et al. U.S. Patent No. 6,321,310 issued November 20, 2001, (hereinafter McCarthy).

Applicant's claimed novel invention, as set forth in representative claim 15, comprises in part:

15. A computer-readable medium, including program instructions executing on a computer, for converting elements of a file access data structure from a first endianness to a second endianness, the method comprising the steps of:

determining if the file access data structure is a critical path data structure;

converting, in response to the file access data structure being a critical path data structure, the elements from the first endianness to the second endianness using a set of specific code functions;

converting, in response to the file access data structure not being a critical path data structure, a header of the file access data structure from the first endianness to the second endianness using a second set of specific code functions; and

calling a byte swapping engine to convert selected elements of the file access data structure from the first byte order to the second byte order.

McCarthy discloses a burst transfers from a memory. A spacing in memory of data words is supplied to a memory controller, and the controller reads out a burst of data words with that spacing.

Applicant respectfully urges that McCarthy has no disclosure of Applicant's claimed use of different processing paths depending upon whether the *file data structure* is a critical path data structure.

Applicant's argument with regard to Claim 2 is applicable to a discussion of claim 13, and is referred to at this point.

Accordingly Applicant respectfully urges that Lee and Bowman-Amuah, and McCarthy taken either singly or in combination, are legally precluded from rendering the presently claimed invention obvious under 35 U.S.C. § 103 because of the absence from both of Applicant's claimed novel:

converting, in response to the file access data structure *being a critical path data structure, the elements from the first endianness to the second endianness using a set of specific code functions;*

converting, in response to the file access data structure not being a critical path data structure, *a header of the file access data structure from the first endianness to the second endianness using a second set of specific code functions*; and  
*calling a byte swapping engine to convert selected elements of the file access data structure from the first byte order to the second byte order.*

At Paragraphs 16-17 of the Office Action, Claims 13 and 14 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bowman-Amuah, Lee, and McCarthy and further in view of Favor U. S. Patent No. 5,926,642 issued July 30, 1999, (hereinafter Favor).

Applicant respectfully notes that claims 13 and 14 are dependent, and are each dependent from an independent claim which is believed to be in condition for allowance. Accordingly, Claims 13 and 14 are believed to be in condition for allowance.

At Paragraphs 18-21 Claim 4 was rejected under 35 U. S. C. § 103 (a) as being unpatentable over Bowman-Amuah, Lee, and McCarthy.

4. A system for converting elements of a file access data structure from a first endianness to a second endianness, the system comprising:  
*an input buffer, the input buffer storing the file access data structure to be converted;*

*a byte swapping engine, the byte swapping engine operative interconnected with a descriptor table; and  
an output buffer, the byte swapping engine placing the file access data structure in the output buffer after conversion.*

Applicant respectfully urges that Bowman-Amuah, Lee, and McCarthy all have no disclosure of Applicant's claimed novel use of *a byte swapping engine, the byte swapping engine operative interconnected with a descriptor table*.

That is, neither Bowman-Amuah, Lee, nor McCarthy have any disclosure of a byte *swapping engine* that is *interconnected with a descriptor table*.

Accordingly, Applicant respectfully urges that Bowman-Amuah, Lee, and McCarthy, taken either singly or in any combination, are legally precluded from rendering the presently claimed invention obvious under 35 U.S.C. § 103 because of the absence from each of Applicant's claimed novel use of *a byte swapping engine, the byte swapping engine operative interconnected with a descriptor table*.

At Paragraph 22 of the Office Action, Claim 5 was rejected as being unpatentable over Bowman-Amuah, Lee, and McCarthy, and further in view of Favor.

Applicant respectfully notes that claim 5 is dependent, and is dependent from an independent claim which is believed to be in condition for allowance. Accordingly, Claim 5 is believed to be in condition for allowance.

At Paragraph 23 of the Office Action, Claim 6 was rejected as being unpatentable over Bowman-Amuah, Lee, and McCarthy, and further in view of Keele.

Applicant respectfully notes that claim 6 is dependent, and is dependent from an independent claim which is believed to be in condition for allowance. Accordingly, Claim 6 is believed to be in condition for allowance.

At Paragraph 24-26 of the Office Action, Claims 7 and 8 were rejected as being unpatentable over Bowman-Amuah and McCarthy.

Applicant's claimed invention, as set forth in representative claim 7, comprises in part:

7. A method for converting a data structure from a first byte order to a second byte order, the method comprising the steps of:  
*reading an element entry from a descriptor table;*  
*performing an action on an element of the data structure, the action being defined in the element entry read from the descriptor table;*  
and  
placing the element in an output buffer.

Applicant respectfully urges that Bowman-Amuah and McCarthy each have no disclosure of Applicant's claimed novel *reading an element entry from a descriptor table;*

*performing an action on an element of the data structure, the action being defined in the element entry read from the descriptor table.*

As mentioned hereinabove, Bowman-Amuah discloses a communication system which uses a client system using a new format to communicate with a legacy system. The new format, for example a web page and a browser to access the web page, contacts a legacy system using an object designed to interface between the new system and the legacy system. The legacy system may be a business oriented software package, and the object provides a communication link between the two systems, the client and the legacy system. The object is designed to specifically provide communication between the particular new system and the particular legacy system.

That is, Bowman-Amuah has no disclosure of Applicant's claimed novel use of a *descriptor table* in performing byte swapping.

Also, McCarthy discloses, as mentioned hereinabove, a burst transfers from a memory. A spacing in memory of data words is supplied to a memory controller, and the controller reads out a burst of data words with that spacing.

Accordingly, McCarthy has no disclosure of Applicant's claimed novel use of a *descriptor table* in performing byte swapping.

Therefore, Applicant respectfully urges that Bowman-Amuah and McCarthy, taken either singly or in combination, are legally precluded from rendering the presently claimed invention obvious under 35 U.S.C. § 103 because of the absence from both of Applicant's claimed novel *reading an element entry from a descriptor table;*

*performing an action on an element of the data structure, the action being defined in the element entry read from the descriptor table.*

At Paragraph 27 of the Office Action, Claim 9 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bowman-Amuah and McCarthy , and further in view of Lee.

Applicant respectfully notes that claim 9 is dependent, and is dependent from an independent claim which is believed to be in condition for allowance. Accordingly, Claim 9 is believed to be in condition for allowance.

At Paragraph 28 of the Office Action, Claim 10 was rejected under 35 U.S.C. § 103 (a) as being unpatentable over Bowman-Amuah and McCarthy , and further in view of Lee.

Applicant respectfully notes that claim 10 is dependent, and is dependent from an independent claim which is believed to be in condition for allowance. Accordingly, Claim 10 is believed to be in condition for allowance.

At Paragraph 29 of the Office Action, Claim 19 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bowman-Amuah, and further in view of Lee.

Applicant respectfully notes that claim 19 is dependent, and is dependent from an independent claim which is believed to be in condition for allowance. Accordingly, Claim 19 is believed to be in condition for allowance.

All independent claims are believed to be in condition for allowance.

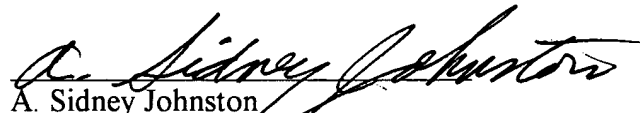
All dependent claims are dependent from independent claims which are believed to be in condition for allowance.

Accordingly, all dependent claims are believed to be in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,



A. Sidney Johnston  
Reg. No. 29,548  
CESARI AND MCKENNA, LLP  
88 Black Falcon Avenue  
Boston, MA 02210-2414  
(617) 951-2500